```
MMM         MMM     OOOOOOOOO     UUU         UUU  NNN         NNN  TTTTTTTTTTTTTTT
MMM         MMM     OOOOOOOOO     UUU         UUU  NNN         NNN  TTTTTTTTTTTTTTT
MMM         MMM     OOOOOOOOO     UUU         UUU  NNN         NNN  TTTTTTTTTTTTTTT
MMMMMM   MMMMMM  OOO         OOO  UUU         UUU  NNN         NNN        TTT
MMMMMM   MMMMMM  OOO         OOO  UUU         UUU  NNN         NNN        TTT
MMMMMM   MMMMMM  OOO         OOO  UUU         UUU  NNN         NNN        TTT
MMM   MMM   MMM  OOO         OOO  UUU         UUU  NNNNNN      NNN        TTT
MMM   MMM   MMM  OOO         OOO  UUU         UUU  NNNNNN      NNN        TTT
MMM   MMM   MMM  OOO         OOO  UUU         UUU  NNNNNN      NNN        TTT
MMM         MMM  OOO         OOO  UUU         UUU  NNN   NNN   NNN        TTT
MMM         MMM  OOO         OOO  UUU         UUU  NNN   NNN   NNN        TTT
MMM         MMM  OOO         OOO  UUU         UUU  NNN   NNN   NNN        TTT
MMM         MMM  OOO         OOO  UUU         UUU  NNN      NNNNNN        TTT
MMM         MMM  OOO         OOO  UUU         UUU  NNN      NNNNNN        TTT
MMM         MMM  OOO         OOO  UUU         UUU  NNN      NNNNNN        TTT
MMM         MMM  OOO         OOO  UUU         UUU  NNN         NNN        TTT
MMM         MMM  OOO         OOO  UUU         UUU  NNN         NNN        TTT
MMM         MMM  OOO         OOO  UUU         UUU  NNN         NNN        TTT
MMM         MMM     OOOOOOOOO     UUUUUUUUUUUUUUU  NNN         NNN        TTT
MMM         MMM     OOOOOOOOO     UUUUUUUUUUUUUUU  NNN         NNN        TTT
MMM         MMM     OOOOOOOOO     UUUUUUUUUUUUUUU  NNN         NNN        TTT
```

```
**FILE**ID**MAKLOG                                        L  9




MM      MM    AAAAAA    KK      KK  LL            OOOOOO      GGGGGGGG
MM      MM    AAAAAA    KK      KK  LL            OOOOOO      GGGGGGGG
MMMM  MMMM  AA      AA  KK      KK  LL          OO      OO  GG
MMMM  MMMM  AA      AA  KK      KK  LL          OO      OO  GG
MM  MM  MM  AA      AA  KK    KK    LL          OO      OO  GG
MM  MM  MM  AA      AA  KK    KK    LL          OO      OO  GG
MM      MM  AA      AA  KKKKKK      LL          OO      OO  GG
MM      MM  AA      AA  KKKKKK      LL          OO      OO  GG
MM      MM  AAAAAAAAAA  KK    KK    LL          OO      OO  GG  GGGGGG
MM      MM  AAAAAAAAAA  KK    KK    LL          OO      OO  GG  GGGGGG
MM      MM  AA      AA  KK      KK  LL          OO      OO  GG      GG      ....
MM      MM  AA      AA  KK      KK  LL          OO      OO  GG      GG      ....
MM      MM  AA      AA  KK      KK  LLLLLLLLLL    OOOOOO      GGGGGG        ....
MM      MM  AA      AA  KK      KK  LLLLLLLLLL    OOOOOO      GGGGGG        ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
0 MODULE MAKLOG (
0                   LANGUAGE (BLISS32),
0                   IDENT = 'V04-000'
0                   ) =
1 BEGIN
1
1 !
1 !****************************************************************************
1 !*                                                                          *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
1 !*  ALL RIGHTS RESERVED.                                                    *
1 !*                                                                          *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
1 !*  TRANSFERRED.                                                            *
1 !*                                                                          *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
1 !*  CORPORATION.                                                            *
1 !*                                                                          *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
1 !*                                                                          *
1 !*                                                                          *
1 !****************************************************************************
1
1 !++
1 !
1 ! FACILITY:  MOUNT Utility Structure Levels 1 & 2
1 !
1 ! ABSTRACT:
1 !
1 !       These routines allocate and hook up the logical name and mounted
1 !       volume list entries.
1 !
1 ! ENVIRONMENT:
1 !
1 !       STARLET operating system, including privileged system services
1 !       and internal exec routines.
1 !
1 !--
1 !
1 !
1 ! AUTHOR:  Andrew C. Goldstein,  CREATION DATE:  20-Oct-1977  19:30
1 !
1 ! MODIFIED BY:
1 !
1 !       V03-018 HH0041          Hai Huang               24-Jul-1984
1 !               Remove REQUIRE 'LIBD$:[VMSLIB.OBJ]MOUNTMSG.B32'.
1 !
1 !       V03-017 HH0040          Hai Huang               20-Jul-1984
1 !               Cal EXE$CRE_GTABLE to create the logical name table if
1 !               it doesn't already exist.
```

```
!
!       V03-016 HH0038          Hai Huang               12-Jul-1984
!               Correct MOUNT_FLAGS structure attribute.
!
!       V03-015 MHB0153         Mark Bramhall           27-Apr-1984
!               Correct NSA$B_ARG_FLAG setting for multiple audits enabled.
!
!       V03-014 ACG0423         Andrew C. Goldstein,    24-Apr-1984  11:06
!               Make disk logical names in MOUNT exec mode to make them
!               available to privileged programs.
!
!       V03-013 HH0012          Hai Huang               09-Apr-1984
!               Get the device owner UIC and the volume protection
!               from the ORB instead of the UCB.
!
!       V03-012 HH0009          Hai Huang               27-Mar-1984
!               Add security auditing support.
!
!       V03-011 HH0007          Hai Huang               21-Mar-1984
!               Add cluster-wide group-volume support, i.e., create the
!               group logical name in the group table of the current
!               process.
!
!       V03-010 HH0002          Hai Huang               23-Jan-1984
!               Add job-wide mount support.
!
!       V03-009 ROW0254         Ralph O. Weber          12-NOV-1983
!               Cause logical names defined herein to be of the GETDVI
!               FULLDEVNAM form.  This will result in allocation class names
!               being used for the equivalence name strings of logical names
!               defined by mount.  Therefore, the mounted volume logical name
!               equivalence strings can be held over time and passed around
!               the VAXcluster without becomming stale.
!
!       V03-008 CDS0001         Christian D. Saether     2-Aug-1983
!               Remove references to RVX structure (obselete).
!
!       V03-007 DMW4057         DMWalp                  23-Jun-1983
!               Change $xxLNM value parameters to be by reference
!
!       V03-006 DMW4050         DMWalp                  15-Jun-1983
!               Corrections to DMW4033, added LNM$M_TERMINAL
!               Change over to LNM$_LNMB_ADDR
!
!       V03-005 ADE9004         A.ELDRIDGE              29-May-1983
!               Fixed name binding to logical name tables.
!
!       V03-004 DMW4033         DMWalp                  26-May-1983
!               Intergate new logical name structures.
!
!       V03-003 STJ50311        Steven T. Jeffreys,     10-Feb-1982
!               - Make all uses of PHYS_NAME indexed by DEVICE_INDEX,
!                 but always use PHYS_NAME[0] for tape mounts.
!               - Set the access mode of the logical names(s) created
!                 to be the MIN (PSL$C_SUPER,.CALLERS_ACMOD).  (SPR 45688)
!
!       V03-002 DMW4010         DMWalp                  19-Nov-1982
```

```
1 !                     Rework logical name block to MTL ( or UCB ) links.
1 !
1 !             V03-001 STJ0248         Steven T. Jeffreys,     31-Mar-1982
1 !                     - Allow for ASCII "A" characters in a volume name.
1 !
1 !             V02-006 STJ0205         Steven T. Jeffreys,     07-Feb-1982
1 !                     Create a local copy of the user specified logical name
1 !                     to prevent it from being stepped on.
1 !
1 !             V02-005 LMP0006         L. Mark Pilant,         29-Dec-1981 12:00
1 !                     Interlock the mount list to avoid potential disasters.
1 !
1 !             V02-004 ACG0219         Andrew C. Goldstein,    23-Oct-1981  10:48
1 !                     Add concealed device support in MOUNT
1 !
1 !             V02-003 STJ0122         Steven T. Jeffreys,     10-Sep-1981
1 !                     Fixed references to the logical name descriptor to use
1 !                     the symbolic offsets.  This ensures that references to
1 !                     the logical name length will be WORD context.
1 !
1 !             V02-002 ACG0167         Andrew C. Goldstein,    18-Apr-1980  13:38
1 !                     Previous revision history moved to MOUNT.REV
1 !**
1
1
1 LIBRARY 'SYS$LIBRARY:LIB.L32';
1 REQUIRE 'SRC$:MOUDEF.B32';
1
1
1 LITERAL
1               PHYS_LENGTH     = 15;           ! longest allowable physical name
1
1 FORWARD ROUTINE
1               LABEL_LENGTH;                   ! return the length of a volume label
```

```
1 GLOBAL ROUTINE ALLOC_LOGNAME (MODE) : NOVALUE =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !        This routine allocates the mounted volume list entry from the
1 !        appropriate storage pools.  It used to allocate logical name block
1 !        also ( thus the name ).
1 !
1 !
1 ! CALLING SEQUENCE:
1 !        ALLOC_LOGNAME ()
1 !
1 ! INPUT PARAMETERS:
1 !        MODE: 0 to use user-specified logical name
1 !              1 to force use of volume name
1 !
1 ! IMPLICIT INPUTS:
1 !        MOUNT parser database
1 !
1 ! OUTPUT PARAMETERS:
1 !        NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !        MTL_ENTRY: address of MTL block
1 !
1 ! ROUTINE VALUE:
1 !        NONE
1 !
1 ! SIDE EFFECTS:
1 !        NONE
1 !
1 !--
1
2 BEGIN
2
2 EXTERNAL
2        MOUNT_OPTIONS   : BITVECTOR,    ! command options
2        MTL_ENTRY       : REF BBLOCK;   ! MTL block
2
2 EXTERNAL ROUTINE
2        ALLOCATE_MEM;                   ! allocate dynamic memory
2
2
2 ! Now allocate the mounted volume list entry.
2 ! Note: to support job-wide mount, a mount list entry
2 ! is always allocated from paged pool.
2 !
2 MTL_ENTRY = ALLOCATE_MEM (MTL$C_LENGTH, 1);
2
2 MTL_ENTRY[MTL$B_TYPE] = DYN$C_MTL;
2
1 END;                                   ! end of routine ALLOC_LOGNAME

                                                        .TITLE  MAKLOG
                                                        .IDENT  \V04-000\

                                                        .EXTRN  MOUNT_OPTIONS, MTL_ENTRY
                                                        .EXTRN  ALLOCATE_MEM

                                                        .PSECT  $CODE$,NOWRT,2

                                     0000 00000         .ENTRY  ALLOC_LOGNAME, Save nothing                 ; 0680
                                  01   DD 00002         PUSHL   #1                                           ; 0729
                                  18   DD 00004         PUSHL   #24                                          ;
                  0000G   CF      02   FB 00006         CALLS   #2, ALLOCATE_MEM                             ;
                  0000G   CF      50   D0 0000B         MOVL    R0, MTL_ENTRY                                ;
                     0A   A0      19   90 00010         MOVB    #25, 10(R0)                                  ; 0731
                                       04 00014         RET                                                  ; 0733

; Routine Size:  21 bytes,    Routine Base:  $CODE$ + 0000
```

```
GLOBAL ROUTINE ENTER_LOGNAME (UCB, VCB) : NOVALUE =

!++
!
! FUNCTIONAL DESCRIPTION:
!
!       This routine completes the logical name and mounted volume list
!       entries.  It builds MTL entry and creates the logical name
!       and hooks up the MTL entry in the appropriate list.
!
!
! CALLING SEQUENCE:
!       ENTER_LOGNAME (ARG1, ARG2)
!
! INPUT PARAMETERS:
!       ARG1: UCB of volume being mounted
!       ARG2: VCB of volume being mounted
!
! IMPLICIT INPUTS:
!       MOUNT parser data base
!       MTL_ENTRY: address of MTL block
!       SMTL_ENTRY: address of MTL block for volume set
!
! OUTPUT PARAMETERS:
!       NONE
!
! IMPLICIT OUTPUTS:
!       NONE
!
! ROUTINE VALUE:
!       NONE
!
! SIDE EFFECTS:
!       logical name and MTL entry entered
!
!--

BEGIN

MAP
        UCB             : REF BBLOCK,   ! UCB being mounted
        VCB             : REF BBLOCK;   ! VCB being mounted

BUILTIN
        INSQUE,
        CALLG;

BIND
        TAPE_PREFIX     = UPLIT BYTE ( 'TAPE$' ),
        DISK_PREFIX     = UPLIT BYTE ( 'DISK$' ),
        SYSTEM_TABLE    = %ASCID 'LNM$SYSTEM',
        JOB_TABLE       = %ASCID 'LNM$JOB';

LOCAL
        ACMODE,                                 ! access mode
        INDEX,                                  ! local index into PHYS_NAME vector
        P,                                      ! string pointer
```

```
2               C,                              ! string count
2               RVT             : REF BBLOCK,   ! pointer to RVT
2               NAME_DESC       : BBLOCK [DSC$K_S_BLN],
2                                               ! internal logical name descriptor
2               LOG_BUFFER      : VECTOR [LNM$C_NAMLENGTH,BYTE],
2                                               ! logical name buffer
2               MOUNT_LIST      : REF BBLOCK,   ! address of mount list tail
2               ITEM_LIST       : VECTOR [(6*3)+1,LONG],
2                                               ! $CRELNM item list, 6 items each
2                                               ! 3 longwords in lenght plus 1
2                                               ! for the terminator longword
2               PHYSNAM_DESC    : BBLOCK [ DSC$K_S_BLN ],
2                                               ! GETDVI descriptor for physical name
2               FULLNAM         : VECTOR [ PHYS_LENGTH + 2, BYTE ],
2                                               ! Place to store the FULLDEVNAM string
2               DVI_ITEM        : VECTOR [ 3+1, LONG ],
2                                               ! GETDVI item list
2               JIB             : REF BBLOCK,   ! pointer to Job Info Block
2               TABLE_NAME      : VECTOR [16, BYTE]
2                                 INITIAL (%ASCII 'LNM$GROUP_000000'),
2                                               ! Group table name
2               GROUP_TABLE     : VECTOR [2, LONG]
2                                 INITIAL (16, TABLE_NAME),
2                                               ! Group table name descriptor
2               ASC_GROUP       : VECTOR [8, BYTE]
2                                 INITIAL (%ASCII '00000000'),
2                                               ! Group in ASCII (6 bytes used)
2               ASC_GROUP_DESC  : VECTOR [2, LONG]
2                                 INITIAL (6, ASC_GROUP);
2                                               ! ASCII group descriptor
2
2       EXTERNAL
2               MOUNT_OPTIONS   : BITVECTOR,    ! command options
2               MOUNT_FLAGS     : VECTOR,       ! mount flags
2               CALLERS_ACMOD   : LONG,         ! Caller's (of $MOUNT) access mode
2               DEVICE_CHAR     : BBLOCK,       ! device characteristics
2               DEVICE_COUNT,                   ! number of devices specified
2               LOG_NAME        : VECTOR,       ! logical name descriptor
2               DEVICE_INDEX    : LONG,         ! index into PHYS_NAME vector
2               PHYS_NAME       : VECTOR,       ! physical device name descriptor
2               MTL_ENTRY       : REF BBLOCK,   ! MTL block
2               SMTL_ENTRY      : REF BBLOCK,   ! MTL block for volume set
2               SCH$GL_CURPCB   : REF BBLOCK ADDRESSING_MODE (GENERAL),
2                                               ! address of our PCB
2               IOC$GQ_MOUNTLST : VECTOR ADDRESSING_MODE (GENERAL),
2                                               ! system mounted volume list head
2               EXE$GL_FLAGS    : BITVECTOR ADDRESSING_MODE (GENERAL),
2                                               ! exec flags longword
2               NSA$GR_ALARMVEC : BBLOCK ADDRESSING_MODE (GENERAL),
2                                               ! alarm enable bit vector
2               NSA$GR_JOURNVEC : BBLOCK ADDRESSING_MODE (GENERAL);
2                                               ! journal enable bit vector
2
2       EXTERNAL LITERAL
2               EXE$V_CONCEALED : UNSIGNED (6); ! concealed device flag
2
```

```
2 LINKAGE
2         ARGLST_IMGNAM   = JSB (REGISTER = 2;) :
2                           NOPRESERVE (0,1)
2                           NOTUSED (3,4,5,6,7,8,9,10,11),
2
2         EXE_CRE_GTABLE  = JSB (REGISTER = 11) :
2                           NOPRESERVE (0,1,2,3,4,5,8);
2
2 EXTERNAL ROUTINE
2         LOCK_IODB,                              ! lock the I/O data base
2         UNLOCK_IODB,                            ! unlock the I/O data base
2         NSA$EVENT_AUDIT : ADDRESSING_MODE (GENERAL),
2                                                 ! security auditing routine
2         NSA$ARGLST_IMGNAM : ARGLST_IMGNAM ADDRESSING_MODE (GENERAL),
2                                                 ! insert IMGNAM into ARGLST
2         EXE$CRE_GTABLE  : EXE_CRE_GTABLE ADDRESSING_MODE (GENERAL);
2                                                 ! create group logical name table
2
2
2 ! First build the volume logical name table entry.
2 ! Use logical name from command unless:
2 !       - There is no logical name
2 !       - It is a disk volume set
2 !       - More than one device is being mounted, and they are not magtapes.
2 !
2 ! Get the logical name; either from the command or from the volume label.
2 !
2
2 ! Copy the user-specified logical name to local storage.
2 !
2
2 CH$MOVE (.LOG_NAME[0], .LOG_NAME[1], LOG_BUFFER);
2 NAME_DESC [DSC$W_LENGTH] = .LOG_NAME [0];
2 NAME_DESC [DSC$B_DTYPE] = 0;
2 NAME_DESC [DSC$B_CLASS] = 0;
2 NAME_DESC [DSC$A_POINTER] = LOG_BUFFER;
2
2 !
2 ! Calculate the access mode for the logical name creation
2 !
2
3 ACMODE = MIN ((IF .MOUNT_OPTIONS[OPT_SYSTEM]
3                 THEN PSL$C_EXEC
3                 ELSE PSL$C_SUPER), .CALLERS_ACMOD);
2
2 IF NOT .MOUNT_OPTIONS[OPT_LOG_NAME]
2 OR .SMTL_ENTRY NEQ 0
3 OR (.DEVICE_COUNT NEQ 1 AND (NOT .DEVICE_CHAR[DEV$V_SQD]))
2 THEN
3     BEGIN
3     IF .DEVICE_CHAR[DEV$V_SQD]
3     THEN P = TAPE_PREFIX
3     ELSE P = DISK_PREFIX;
3
3     C = LABEL_LENGTH (VCB$S_VOLNAME, VCB[VCB$T_VOLNAME]);
3     NAME_DESC[DSC$W_LENGTH] = .C + 5;
3     NAME_DESC[DSC$A_POINTER] = LOG_BUFFER;
```

```
3     CH$COPY (5, .P, .C, VCB[VCB$T_VOLNAME], 0, .C+5, LOG_BUFFER);
2     END;
2
2 ! Now create logical name.  The physical device string is the equivalence
2 ! string.  If a tape mount, use the physical name of the first volume,
2 ! otherwise use the physical name of the current volume.
2 !
2
2 INDEX = .DEVICE_INDEX;
2 IF .BBLOCK [UCB[UCB$L_DEVCHAR], DEV$V_SQD]
2 THEN
2     INDEX = 0;
2
2 ! Store the location of the LNM block in the MTL
2 !
2 ITEM_LIST [ 0 ] = ( LNM$_LNMB_ADDR^16 OR 4 );
2 ITEM_LIST [ 1 ] = MTL_ENTRY[MTL$L_LOGNAME]; ! CAUTION USED BY ITEM_LIST [ 7 ]
2 ITEM_LIST [ 2 ] = 0;
2
2 ! Store the location of the MTL in the LNM BLOCK.
2 ! This causes the logical name deletion logic to clear the MTL's logical name
2 ! pointer if the logical name is deleted, just as it does when a mailbox
2 ! logical name is deleted.
2 !
2 ITEM_LIST [ 3 ] = ( LNM$_INDEX^16 or 4 );
2 ITEM_LIST [ 4 ] = UPLIT ( LNMX$C_BACKPTR );
2 ITEM_LIST [ 5 ] = 0;
2 ITEM_LIST [ 6 ] = ( LNM$_STRING^16 or 4 );
2 ITEM_LIST [ 7 ] = ITEM_LIST [ 1 ];
2 ITEM_LIST [ 8 ] = 0;
2
2 ! Define equivalence string
2 !
2 ITEM_LIST [  9 ] = ( LNM$_INDEX^16 or 4 );
2 ITEM_LIST [ 10 ] = UPLIT ( 0 );
2 ITEM_LIST [ 11 ] = 0;
2
2 ITEM_LIST [ 12 ] = ( LNM$_ATTRIBUTES^16 or 4 );
3 ITEM_LIST [ 13 ] = ( IF .EXE$GL_FLAGS[EXE$V_CONCEALED]
3                     THEN UPLIT ( LNM$M_TERMINAL OR LNM$M_CONCEALED )
2                     ELSE UPLIT ( LNM$M_TERMINAL ) );
2 ITEM_LIST [ 14 ] = 0;
2
2 ! Use GETDVI to obtain the most universal device name for this physical
2 ! device, FULLDEVNAM, and pass that to CRELNM as the equivalence name
2 ! string.
2 !
2 PHYSNAM_DESC [ DSC$W_LENGTH ]  = .PHYS_NAME [ .INDEX*2 ] - 1;
2 PHYSNAM_DESC [ DSC$A_POINTER ] = .PHYS_NAME [ .INDEX*2 + 1 ] + 1;
2 PHYSNAM_DESC [ DSC$B_DTYPE ] = 0;
2 PHYSNAM_DESC [ DSC$B_CLASS ] = 0;
2
2 DVI_ITEM [ 0 ] = ( DVI$_FULLDEVNAM^16 or ( PHYS_LENGTH + 2 ) );
2 DVI_ITEM [ 1 ] = FULLNAM;
2 DVI_ITEM [ 2 ] = ITEM_LIST [ 15 ];
2 DVI_ITEM [ 3 ] = 0;
2 ITEM_LIST [ 15 ] = 0;
```

```
  2
P 2 $GETDVIW (
P 2         devnam = PHYSNAM_DESC,
  2         itmlst = DVI_ITEM        );
  2
  2 IF .FULLNAM [ 0 ] eql %C'_'
  2     THEN BEGIN
  3         ITEM_LIST [ 15 ] = ( LNM$_STRING^16 or ( .ITEM_LIST [ 15 ] - 1 ) );
  3         ITEM_LIST [ 16 ] = FULLNAM + 1;
  3         END
  2     ELSE BEGIN
  3         ITEM_LIST [ 15 ] = ( LNM$_STRING^16 or .ITEM_LIST [ 15 ] );
  3         ITEM_LIST [ 16 ] = FULLNAM;
  3         END;
  2 ITEM_LIST [ 17 ] = 0;
  2
  2 ! End item list
  2 !
  2 ITEM_LIST [ 18 ] = 0;
  2
  2 !
  2 ! If the volume is to be mounted /group, then we have to create the group logical
  2 ! name in the group of the current process.  To avoid the situation that the group
  2 ! table does not exist, we call the EXE$CRE_GTABLE routine, which creates the group
  2 ! table if it doesn't already exist.
  2 !
  2 IF .MOUNT_OPTIONS [OPT_GROUP]
  2 THEN
  3     BEGIN
  3
P 3     $FAO ( %ASCID 'LNM$GROUP_!OW',                 ! Format LNM$GROUP_xxxxxx
P 3            GROUP_TABLE,
P 3            GROUP_TABLE,
  3            .(SCH$GL_CURPCB [PCB$L_UIC]) <16,16>); ! Convert our group number to octal
  3
P 3     $FAO ( %ASCID '!OW',                           ! Format octal in ASCII
P 3            ASC_GROUP_DESC,
P 3            ASC_GROUP_DESC,
  3            .(SCH$GL_CURPCB [PCB$L_UIC]) <16,16>); ! Convert our group number to octal
  3
  3     EXE$CRE_GTABLE (ASC_GROUP);                     ! Create the LNM$GROUP_xxxxxx table
  3
  2     END;                                            ! exists
  2
  2
P 2 $CRELNM
P 2         ( ACMODE = ACMODE,
P 2           TABNAM = (IF  .MOUNT_OPTIONS [ OPT_SYSTEM ]
P 2                     THEN SYSTEM_TABLE
P 2                     ELSE
P 2                          IF  .MOUNT_OPTIONS [ OPT_GROUP ]
P 2                          THEN GROUP_TABLE
P 2                          ELSE JOB_TABLE
P 2                    ),
P 2           LOGNAM = NAME_DESC,
  2           ITMLST = ITEM_LIST );
  2
```

```
2 ! Link the MTL entry into the list
2 !
2 MTL_ENTRY[MTL$L_UCB] = .UCB;
2 LOCK_IODB ();                                 ! lock the mount list
2
2 IF .MOUNT_OPTIONS[OPT_GROUP] OR .MOUNT_OPTIONS[OPT_SYSTEM]
2 THEN MOUNT_LIST = IOC$GQ_MOUNTLST[1]
2 ELSE
3     BEGIN
3     JIB = .SCH$GL_CURPCB[PCB$L_JIB];
3     MOUNT_LIST = JIB[JIB$L_MTLBL];            ! get the tail of the mount list
2     END;
2 INSQUE (.MTL_ENTRY, ..MOUNT_LIST);
2
2 UNLOCK_IODB ();                               ! unlock the mount list
2
2 ! Now build the volume set logical name if we are mounting volume 1 of a
2 ! disk volume set.
2 !
2 IF .SMTL_ENTRY NEQ 0
2 THEN
3     BEGIN
3
3     ! Get the logical name; either from the command or from the volume label.
3     !
3
3     ! Copy the user-specified logical name to local storage.
3     !
3     CH$MOVE (.LOG_NAME[0], .LOG_NAME[1], LOG_BUFFER);
3     NAME_DESC [DSC$W_LENGTH] = .LOG_NAME [0];
3     NAME_DESC [DSC$B_DTYPE] = 0;
3     NAME_DESC [DSC$B_CLASS] = 0;
3     NAME_DESC [DSC$A_POINTER] = LOG_BUFFER;
3
3     IF NOT .MOUNT_OPTIONS[OPT_LOG_NAME]
3     THEN
4         BEGIN
4         IF .DEVICE_CHAR[DEV$V_SQD]
4         THEN P = TAPE_PREFIX
4         ELSE P = DISK_PREFIX;
4
4         RVT = .VCB[VCB$L_RVT];
4         C = LABEL_LENGTH (RVT$S_STRUCNAME, RVT[RVT$T_STRUCNAME]);
4         NAME_DESC[DSC$W_LENGTH] = .C + 5;
4         NAME_DESC[DSC$A_POINTER] = LOG_BUFFER;
4         CH$COPY (5, .P, .C, RVT[RVT$T_STRUCNAME], 0, .C+5, LOG_BUFFER);
3         END;
3
3     ! Now create logical name.  The physical device string is the equivalence
3     ! string.  If a tape mount, use the physical name of the first volume,
3     ! otherwise use the physical name of the current volume.
3     !
3     INDEX = .DEVICE_INDEX;
3     IF .BBLOCK [UCB[UCB$L_DEVCHAR], DEV$V_SQD]
3     THEN
3         INDEX = 0;
3
```

```
  3        ! Store the location of the LNM block in the MTL
  3        !
  3        ITEM_LIST [ 0 ] = ( LNM$_LNMB_ADDR^16 OR 4 );
  3        ITEM_LIST [ 1 ] = $MTL_ENTRY[MTL$L_LOGNAME];
  3        ITEM_LIST [ 2 ] = 0;
  3
  3        ! Store the location of the MTL in the LNM BLOCK.
  3        ! This causes the logical name deletion logic to clear the MTL's logical
  3        ! name pointer if the logical name is deleted, just as it does when a
  3        ! mailbox logical name is deleted.
  3
  3        ITEM_LIST [ 3 ] = ( LNM$_INDEX^16 or 4 );
  3        ITEM_LIST [ 4 ] = UPLIT ( LNMX$C_BACKPTR );
  3        ITEM_LIST [ 5 ] = 0;
  3        ITEM_LIST [ 6 ] = ( LNM$_STRING^16 or 4 );
  3        ITEM_LIST [ 7 ] = ITEM_LIST [ 1 ];
  3        ITEM_LIST [ 8 ] = 0;
  3
  3        ! Define equivalence string
  3        !
  3        ITEM_LIST [  9 ] = ( LNM$_INDEX^16 or 4 );
  3        ITEM_LIST [ 10 ] = UPLIT ( 0 );
  3        ITEM_LIST [ 11 ] = 0;
  3
  3        ITEM_LIST [ 12 ] = ( LNM$_ATTRIBUTES^16 or 4 );
  4        ITEM_LIST [ 13 ] = ( IF .EXE$GL_FLAGS[EXE$V_CONCEALED]
  4                             THEN UPLIT ( LNM$M_TERMINAL OR LNM$M_CONCEALED )
  3                             ELSE UPLIT ( LNM$M_TERMINAL ) );
  3        ITEM_LIST [ 14 ] = 0;
  3
  3    ! Use GETDVI to obtain the most universal device name for this physical
  3    ! device, FULLDEVNAM, and pass that to CRELNM as the equivalence name
  3    ! string.
  3    !
  3    PHYSNAM_DESC [ DSC$W_LENGTH ]  = .PHYS_NAME [ .INDEX*2 ] - 1;
  3    PHYSNAM_DESC [ DSC$A_POINTER ] = .PHYS_NAME [ .INDEX*2 + 1 ] + 1;
  3    PHYSNAM_DESC [ DSC$B_DTYPE ] = 0;
  3    PHYSNAM_DESC [ DSC$B_CLASS ] = 0;
  3
  3    DVI_ITEM [ 0 ] = ( DVI$_FULLDEVNAM^16 or ( PHYS_LENGTH + 2 ) );
  3    DVI_ITEM [ 1 ] = FULLNAM;
  3    DVI_ITEM [ 2 ] = ITEM_LIST [ 15 ];
  3    DVI_ITEM [ 3 ] = 0;
  3    ITEM_LIST [ 15 ] = 0;
  3
P 3    $GETDVIW (
P 3            devnam = PHYSNAM_DESC,
  3            itmlst = DVI_ITEM        );
  3
  3    IF .FULLNAM [ 0 ] eql %C'_'
  4        THEN BEGIN
  4            ITEM_LIST [ 15 ] = ( LNM$_STRING^16 or ( .ITEM_LIST [ 15 ] - 1 ) );
  4            ITEM_LIST [ 16 ] = FULLNAM + 1;
  4            END
  4        ELSE BEGIN
  4            ITEM_LIST [ 15 ] = ( LNM$_STRING^16 or .ITEM_LIST [ 15 ] );
  4            ITEM_LIST [ 16 ] = FULLNAM;
```

```
        END;
    ITEM_LIST [ 17 ] = 0;

    ! End item list
    !
    ITEM_LIST [ 18 ] = 0;

    $CRELNM
            ( ACMODE = ACMODE,
              TABNAM = (IF   .MOUNT_OPTIONS [ OPT_SYSTEM ]
                        THEN SYSTEM_TABLE
                        ELSE
                            IF   .MOUNT_OPTIONS [ OPT_GROUP ]
                            THEN GROUP_TABLE
                            ELSE JOB_TABLE
                       ),
              LOGNAM = NAME_DESC,
              ITMLST = ITEM_LIST );

    SMTL_ENTRY[MTL$L_UCB] = .UCB;
    SMTL_ENTRY[MTL$V_VOLSET] = 1;        ! identify as a volume set entry

    LOCK_IODB ();                        ! lock the mount list

    IF .MOUNT_OPTIONS[OPT_GROUP] OR .MOUNT_OPTIONS[OPT_SYSTEM]
    THEN MOUNT_LIST = IOC$GQ_MOUNTLST[1]
    ELSE
        BEGIN
        JIB = .SCH$GL_CURPCB[PCB$L_JIB];
        MOUNT_LIST = JIB[JIB$L_MTLBL];  ! get the tail of the mount list
        END;
    INSQUE (.SMTL_ENTRY, ..MOUNT_LIST);

    UNLOCK_IODB ();                      ! unlock the mount list

    END;


IF (.SCH$GL_CURPCB [PCB$V_SECAUDIT]
OR  .NSA$GR_ALARMVEC [NSA$V_EVT_MOUNT]
OR  .NSA$GR_JOURNVEC [NSA$V_EVT_MOUNT])
THEN
    BEGIN

    LOCAL
        ARGLIST : BBLOCK[NSA$K_ARG2_LENGTH],    ! security auditing argument list
        ORB     : REF BBLOCK,                   ! address of the ORB
        TEMP_PROT;                              ! temporary protection word

    CH$FILL (0, NSA$K_ARG2_LENGTH, ARGLIST);    ! zero argument list
    ORB = .UCB [UCB$L_ORB];                     ! get ORB address

    !
    ! Set up the security auditing argument list header
    !

    ARGLIST [NSA$L_ARG_COUNT] = ( NSA$K_ARG2_LENGTH/4 ) - 4;
```

```
                                                        ! initialize length of argument list
                                                        ! less vol-set pkt and arg count
    ARGLIST [NSA$L_ARG_ID] = NSA$K_RECID_VOL_MOU;
                                                        ! initialize record id as mount
    IF .SCH$GL_CURPCB [PCB$V_SECAUDIT]                  ! set up proper flags
    THEN
        ARGLIST [NSA$V_ARG_FLAG_MANDY] = 1;             ! mandatory auditing
    IF .NSA$GR_ALARMVEC [NSA$V_EVT_MOUNT]
    THEN
        ARGLIST [NSA$V_ARG_FLAG_ALARM] = 1;             ! generate alarm for this record
    IF .NSA$GR_JOURNVEC [NSA$V_EVT_MOUNT]
    THEN
        ARGLIST [NSA$V_ARG_FLAG_JOURN] = 1;             ! journal this record

    ARGLIST [NSA$B_ARG_PKTNUM] = 7;                     ! initialize number of items
                                                        ! less vol-set pkt

    !
    ! Set up the security auditing argument list for mount
    !

    ARGLIST [NSA$L_ARG2_UIC_TM] = NSA$K_ARG_MECH_LONG^16 + NSA$K_PKTTYP_UIC;
    ARGLIST [NSA$L_ARG2_UIC] = .ORB [ORB$L_OWNER];      ! set device owner UIC

    ARGLIST [NSA$L_ARG2_VOLPRO_TM] = NSA$K_ARG_MECH_WORD^16 + NSA$K_PKTTYP_VOLPRO;
    !
    ! Get the volume protection
    !
    TEMP_PROT = 0;                                              ! clear temp location
    IF .ORB [ORB$V_PROT_16]
    THEN
        TEMP_PROT = .ORB [ORB$W_PROT]                           ! standard SOGW protection
    ELSE
        BEGIN                                                   ! vector protection
        TEMP_PROT <0,4>  = .(ORB [ORB$L_SYS_PROT])<0,4>;        ! system
        TEMP_PROT <4,4>  = .(ORB [ORB$L_OWN_PROT])<0,4>;        ! owner
        TEMP_PROT <8,4>  = .(ORB [ORB$L_GRP_PROT])<0,4>;        ! group
        TEMP_PROT <12,4> = .(ORB [ORB$L_WOR_PROT])<0,4>;        ! world
        END;
    ARGLIST [NSA$L_ARG2_VOLPRO] = .TEMP_PROT;                   ! set volume protection mask

    ARGLIST [NSA$L_ARG2_MOUFLG_TM] = NSA$K_ARG_MECH_LONG^16 + NSA$K_PKTTYP_MOUFLG;
    ARGLIST [NSA$L_ARG2_MOUFLG ] = .MOUNT_FLAGS;                ! set mount flags

    NSA$ARGLST_IMGNAM (ARGLIST [NSA$L_ARG2_IMGNAM_TM]); ! set image name

    ARGLIST [NSA$L_ARG2_DEVNAM_TM] = NSA$K_ARG_MECH_DESCR^16 + NSA$K_PKTTYP_DEVNAM;
    IF .FULLNAM [0] EQL %C'_'
    THEN
        ITEM_LIST [15] = .ITEM_LIST [15] + 1;                   ! include the '_' char
    ARGLIST [NSA$L_ARG2_DEVNAM_SIZ] = .ITEM_LIST [15];          ! set size of full device name
    ARGLIST [NSA$L_ARG2_DEVNAM_PTR] = FULLNAM;                  ! set full device name buffer address

    ARGLIST [NSA$L_ARG2_LOGNAM_TM] = NSA$K_ARG_MECH_DESCR^16 + NSA$K_PKTTYP_LOGNAM;
    ARGLIST [NSA$L_ARG2_LOGNAM_SIZ] = .NAME_DESC [DSC$W_LENGTH]; ! set size of logical name
    ARGLIST [NSA$L_ARG2_LOGNAM_PTR] = LOG_BUFFER;               ! set logical name buffer address
```

```
3         ARGLIST [NSA$L_ARG2_VOLNAM_TM] = NSA$K_ARG_MECH_DESCR^16 + NSA$K_PKTTYP_VOLNAM;
3         ARGLIST [NSA$L_ARG2_VOLNAM_SIZ] =
3             LABEL_LENGTH (VCB$S_VOLNAME, VCB [VCB$T_VOLNAME]);  ! set size of volume name
3         ARGLIST [NSA$L_ARG2_VOLNAM_PTR] = VCB [VCB$T_VOLNAME];  ! set volume name buffer address
3
3         !
3         ! If the volume is a member of a volume set, then
3         !     a. increment argument count
3         !     b. increment number of packets
3         !     c. set up volume set descriptor
3         !
3
4         IF ( NOT .BBLOCK [UCB [UCB$L_DEVCHAR], DEV$V_FOR] )
4         AND    ( .VCB [VCB$W_RVN] NEQ 0 )
3         THEN
4             BEGIN
4             ARGLIST [NSA$L_ARG_COUNT] = .ARGLIST [NSA$L_ARG_COUNT] + 3; ! count vol-set pkt
4             ARGLIST [NSA$B_ARG_PKTNUM] = .ARGLIST [NSA$B_ARG_PKTNUM] + 1;
4             ARGLIST [NSA$L_ARG2_VOLSNAM_TM] = NSA$K_ARG_MECH_DESCR^16 + NSA$K_PKTTYP_VOLSNAM;
4             RVT = .VCB [VCB$L_RVT];
4             ARGLIST [NSA$L_ARG2_VOLSNAM_SIZ] =
4                 LABEL_LENGTH (RVT$S_STRUCNAME, RVT [RVT$T_STRUCNAME]); ! set size of vol-set name
4             ARGLIST [NSA$L_ARG2_VOLSNAM_PTR] = RVT [RVT$T_STRUCNAME];  ! set vol-set name buffer address
3             END;
3
3         CALLG (ARGLIST, NSA$EVENT_AUDIT);   ! call event audit routine
3
2       END;                                  ! end of block defining ARGLIST
2
1 END;                                        ! end of routine ENTER_LOGNAME


                                                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

                                24  45  50  41  54  00000 P.AAA:   .ASCII  \TAPE$\
                                24  4B  53  49  44  00005 P.AAB:   .ASCII  \DISK$\
                                                    0000A          .BLKB   2
        00  00  4D  45  54  53  59  53  24  4D  4E  4C  0000C P.AAD:   .ASCII  \LNM$SYSTEM\<0><0>
                                            010E000A  00018 P.AAC:   .LONG   17694730
                                           00000000'  0001C          .ADDRESS P.AAD
                        00  42  4F  4A  24  4D  4E  4C  00020 P.AAF:   .ASCII  \LNM$JOB\<0>
                                            010F0007  00028 P.AAE:   .LONG   17694727
                                           00000000'  0002C          .ADDRESS P.AAF
30  30  30  30  30  5F  50  55  4F  52  47  24  4D  4E  4C  00030 P.AAG:   .ASCII  \LNM$GROUP_000000\
                                                  30  0003F
                                            FFFFFF81  00040 P.AAH:   .LONG   -127
                                            00000000  00044 P.AAI:   .LONG   0
                                            00000300  00048 P.AAJ:   .LONG   768
                                            00000200  0004C P.AAK:   .LONG   512
00  00  57  4F  21  5F  50  55  4F  52  47  24  4D  4E  4C  00050 P.AAM:   .ASCII  \LNM$GROUP_!OW\<0><0><0>
                                                  00  0005F
                                            010E000D  00060 P.AAL:   .LONG   17694733
                                           00000000'  00064          .ADDRESS P.AAM
                                    00  57  4F  21  00068 P.AAO:   .ASCII  \!OW\<0>
                                            010E0003  0006C P.AAN:   .LONG   17694723
                                           00000000'  00070          .ADDRESS P.AAO
```

```
                                 FFFFFF81  00074 P.AAP:  .LONG   -127                                   :
                                 00000000  00078 P.AAQ:  .LONG   0                                      :
                                 00000300  0007C P.AAR:  .LONG   768                                    :
                                 00000200  00080 P.AAS:  .LONG   512                                    :

                                                 TAPE_PREFIX=         P.AAA
                                                 DISK_PREFIX=         P.AAB
                                                 SYSTEM_TABLE=        P.AAC
                                                 JOB_TABLE=           P.AAE
                                                         .EXTRN  MOUNT_FLAGS, CALLERS_ACMOD
                                                         .EXTRN  DEVICE_CHAR, DEVICE_COUNT
                                                         .EXTRN  LOG_NAME, DEVICE_INDEX
                                                         .EXTRN  PHYS_NAME, SMTL_ENTRY
                                                         .EXTRN  SCH$GL_CURPCB, IOC$GQ_MOUNTLST
                                                         .EXTRN  EXE$GL_FLAGS, NSA$GR_ALARMVEC
                                                         .EXTRN  NSA$GR_JOURNVEC
                                                         .EXTRN  EXE$V_CONCEALED
                                                         .EXTRN  LOCK_IODB, UNLOCK_IODB
                                                         .EXTRN  NSA$EVENT_AUDIT
                                                         .EXTRN  NSA$ARGLST_IMGNAM
                                                         .EXTRN  EXE$CRE_GTABLE, SYS$GETDVIW
                                                         .EXTRN  SYS$FAO, SYS$CRELNM

                                                         .PSECT  $CODE$,NOWRT,2

                                      0FFC 00000         .ENTRY  ENTER_LOGNAME, Save R2,R3,R4,R5,R6,R7,R8,-  : 0734
                                                                 R9,R10,R11                                   :
                     5E      FDEC   CE  9E 00002         MOVAB   -532(SP), SP                                 :
0084   CE   0000'    CF             10  28 00007         MOVC3   #16, P.AAG, TABLE_NAME                       : 0810
              7C     AE             10  D0 0000F         MOVL    #16, GROUP_TABLE                             :
            0080     CE      0084   CE  9E 00013         MOVAB   TABLE_NAME, GROUP_TABLE+4                    :
              74     AE  30303030   8F  D0 0001A         MOVL    #808464432, ASC_GROUP                        :
              78     AE  30303030   8F  D0 00022         MOVL    #808464432, ASC_GROUP+4                      :
              6C     AE             06  D0 0002A         MOVL    #6, ASC_GROUP_DESC                           :
              70     AE        74   AE  9E 0002E         MOVAB   ASC_GROUP, ASC_GROUP_DESC+4                  :
FEF8   CD   0000G    DF     0000G   CF  28 00033         MOVC3   LOG_NAME, @LOG_NAME+4, LOG_BUFFER            : 0879
              F8     AD     0000G   CF  3C 0003D         MOVZWL  LOG_NAME, NAME_DESC                          : 0880
              FC     AD      FEF8   CD  9E 00043         MOVAB   LOG_BUFFER, NAME_DESC+4                      : 0883
              05     0000G          CF  E9 00049         BLBC    MOUNT_OPTIONS+1, 1$                          : 0889
              50                    01  D0 0004E         MOVL    #1, R0                                       :
                                    03  11 00051         BRB     2$                                           :
              50                    02  D0 00053 1$:     MOVL    #2, R0                                       :
            0000G    CF             50  D1 00056 2$:     CMPL    R0, CALLERS_ACMOD                            : 0891
                                    05  15 0005B         BLEQ    3$                                           :
              50            0000G   CF  D0 0005D         MOVL    CALLERS_ACMOD, R0                            :
              08     AE             50  D0 00062 3$:     MOVL    R0, ACMODE                                   : 0889
       13   0000G    CF             05  E1 00066         BBC     #5, MOUNT_OPTIONS+3, 4$                      : 0893
                            0000G   CF  D5 0006C         TSTL    SMTL_ENTRY                                   : 0894
                                    0D  12 00070         BNEQ    4$                                           :
              01            0000G   CF  D1 00072         CMPL    DEVICE_COUNT, #1                             : 0895
                                    58  13 00077         BEQL    7$                                           :
       52   0000G    CF             05  E0 00079         BBS     #5, DEVICE_CHAR, 7$                          :
       07   0000G    CF             05  E1 0007F 4$:     BBC     #5, DEVICE_CHAR, 5$                          : 0898
              6E            0000'   CF  9E 00085         MOVAB   TAPE_PREFIX, P                               : 0899
                                    05  11 0008A         BRB     6$                                           :
              6E            0000'   CF  9E 0008C 5$:     MOVAB   DISK_PREFIX, P                               : 0900
              56              08    AC  D0 00091 6$:     MOVL    VCB, R6                                      : 0902
```

```
                                                                  C 11
MAKLOG                                                           16-Sep-1984 01:16:19     VAX-11 Bliss-32 V4.0-742                 Page  17
V04-000                                                          14-Sep-1984 12:45:22     DISK$VMSMASTER:[MOUNT.SRC]MAKLOG.B32;1    (3)

                                                 14   A6  9F 00095         PUSHAB  20(R6)                                      ;
                                                      0C  DD 00098         PUSHL   #12                                         ;
                                0000V  CF             02  FB 0009A         CALLS   #2, LABEL_LENGTH                            ;
                                   04  AE             50  D0 0009F         MOVL    R0, C                                       ;
                         50        04  AE             05  C1 000A3         ADDL3   #5, C, R0                                   ; 0903
                                   F8  AD             50  B0 000A8         MOVW    R0, NAME_DESC                               ;
                                   FC  AD      FEF8   CD  9E 000AC         MOVAB   LOG_BUFFER, NAME_DESC+4                     ; 0904
                                       5A             50  D0 000B2         MOVL    R0, R10                                     ; 0905
                                       57      FEF8   CD  9E 000B5         MOVAB   LOG_BUFFER, R7                              ;
           5A            00        00  BE             05  2C 000BA         MOVC5   #5, @P, #0, R10, (R7)                       ;
                                                      67     000C0                                                             ;
                                                      0E  18 000C1         BGEQ    7$                                          ;
                                       57             05  C0 000C3         ADDL2   #5, R7                                      ;
                                       5A             05  C2 000C6         SUBL2   #5, R10                                     ;
           5A            00        14  A6        04   AE  2C 000C9         MOVC5   C, 20(R6), #0, R10, (R7)                    ;
                                                      67     000D0                                                             ;
                                       56      0000G  CF  D0 000D1 7$:     MOVL    DEVICE_INDEX, INDEX                         ; 0913
                                       57        04   AC  D0 000D6         MOVL    UCB, R7                                     ; 0914
                         02        38  A7             05  E1 000DA         BBC     #5, 56(R7), 8$                              ;
                                                      56  D4 000DF         CLRL    INDEX                                       ; 0916
                                00C0   CE 00090004    8F  D0 000E1 8$:     MOVL    #589828, ITEM_LIST                          ; 0920
                  00C4   CE     0000G  CF             10  C1 000EA         ADDL3   #16, MTL_ENTRY, ITEM_LIST+4                 ; 0921
                                               00C8   CE  D4 000F2         CLRL    ITEM_LIST+8                                 ; 0922
                                00CC   CE 00010004    8F  D0 000F6         MOVL    #65540, ITEM_LIST+12                        ; 0929
                                00D0   CE      0000'  CF  9E 000FF         MOVAB   P.AAH, ITEM_LIST+16                         ; 0930
                                               00D4   CE  D4 00106         CLRL    ITEM_LIST+20                                ; 0931
                                00D8   CE 00020004    8F  D0 0010A         MOVL    #131076, ITEM_LIST+24                       ; 0932
                                00DC   CE      00C4   CE  9E 00113         MOVAB   ITEM_LIST+4, ITEM_LIST+28                   ; 0933
                                               00E0   CE  D4 0011A         CLRL    ITEM_LIST+32                                ; 0934
                                00E4   CE 00010004    8F  D0 0011E         MOVL    #65540, ITEM_LIST+36                        ; 0938
                                00E8   CE      0000'  CF  9E 00127         MOVAB   P.AAI, ITEM_LIST+40                         ; 0939
                                               00EC   CE  D4 0012E         CLRL    ITEM_LIST+44                                ; 0940
                                00F0   CE 00030004    8F  D0 00132         MOVL    #196612, ITEM_LIST+48                       ; 0942
                         07 00000000G  00             00G E1 0013B         BBC     S^EXE$V_CONCEALED, EXE$GL_FLAGS, 9$         ; 0943
                                       50      0000'  CF  9E 00143         MOVAB   P.AAJ, R0                                   ; 0944
                                                      05  11 00148         BRB     10$                                         ;
                                       50      0000'  CF  9E 0014A 9$:     MOVAB   P.AAK, R0                                   ; 0945
                                00F4   CE             50  D0 0014F 10$:    MOVL    R0, ITEM_LIST+52                            ; 0943
                                               00F8   CE  D4 00154         CLRL    ITEM_LIST+56                                ; 0946
                         50            56             01  78 00158         ASHL    #1, INDEX, R0                               ; 0952
                                               0000GCF40  DF 0015C         PUSHAL  PHYS_NAME[R0]                               ;
                  00B8   CE            9E             01  A3 00161         SUBW3   #1, @(SP)+, PHYSNAM_DESC                    ;
                  00BC   CE     0000GCF40             01  C1 00167         ADDL3   #1, PHYS_NAME+4[R0], PHYSNAM_DESC+4         ; 0953
                                               00BA   CE  B4 00170         CLRW    PHYSNAM_DESC+2                              ; 0954
                                0094   CE 00E80011    8F  D0 00174         MOVL    #15204369, DVI_ITEM                         ; 0957
                                0098   CE      00A4   CE  9E 0017D         MOVAB   FULLNAM, DVI_ITEM+4                         ; 0958
                                009C   CE      00FC   CE  9E 00184         MOVAB   ITEM_LIST+60, DVI_ITEM+8                    ; 0959
                                               00A0   CE  D4 0018B         CLRL    DVI_ITEM+12                                 ; 0960
                                               00FC   CE  D4 0018F         CLRL    ITEM_LIST+60                                ; 0961
                                                      7E  7C 00193         CLRQ    -(SP)                                       ; 0965
                                                      7E  7C 00195         CLRQ    -(SP)                                       ;
                                               00A4   CE  9F 00197         PUSHAB  DVI_ITEM                                    ;
                                               00CC   CE  9F 0019B         PUSHAB  PHYSNAM_DESC                                ;
                                                      7E  7C 0019F         CLRQ    -(SP)                                       ;
                            00000000G  00             08  FB 001A1         CALLS   #8, SYS$GETDVIW                             ;
                                   5F  8F      00A4   CE  91 001A8         CMPB    FULLNAM, #95                                ; 0967
                                                      19  12 001AE         BNEQ    11$                                         ;
```

```
                50      00FC    CE              01  C3 001B0        SUBL3   #1, ITEM_LIST+60, R0                         ; 0969
      00FC      CE              50 00020000     8F  C9 001B6        BISL3   #131072, R0, ITEM_LIST+60
                        0100    CE       00A5   CE  9E 001C0        MOVAB   FULLNAM+1, ITEM_LIST+64                      ; 0970
                                                0C  11 001C7        BRB     12$                                          ; 0967
                        00FE    CE              02  88 001C9 11$:   BISB2   #2, ITEM_LIST+60                             ; 0973
                        0100    CE       00A4   CE  9E 001CE        MOVAB   FULLNAM, ITEM_LIST+64                        ; 0974
                                         0104   CE  7C 001D5 12$:   CLRQ    ITEM_LIST+68                                 ; 0976
                                        0000G   CF  95 001D9        TSTB    MOUNT_OPTIONS                                ; 0988
                                                46  18 001DD        BGEQ    13$
                                50 00000000G    00  D0 001DF        MOVL    SCH$GL_CURPCB, R0                            ; 0995
                                7E       00BE   C0  3C 001E6        MOVZWL  190(R0), -(SP)
                                         0080   CE  9F 001EB        PUSHAB  GROUP_TABLE
                                         0084   CE  9F 001EF        PUSHAB  GROUP_TABLE
                                        0000'   CF  9F 001F3        PUSHAB  P.AAL
                    00000000G   00              04  FB 001F7        CALLS   #4, SYS$FAO
                                50 00000000G    00  D0 001FE        MOVL    SCH$GL_CURPCB, R0                            ; 1000
                                7E       00BE   C0  3C 00205        MOVZWL  190(R0), -(SP)
                                           70   AE  9F 0020A        PUSHAB  ASC_GROUP_DESC
                                           74   AE  9F 0020D        PUSHAB  ASC_GROUP_DESC
                                        0000'   CF  9F 00210        PUSHAB  P.AAN
                    00000000G   00              04  FB 00214        CALLS   #4, SYS$FAO
                                5B         74   AE  9E 0021B        MOVAB   ASC_GROUP, R11                               ; 1002
                                   00000000G    00  16 0021F        JSB     EXE$CRE_GTABLE
                                         00C0   CE  9F 00225 13$:   PUSHAB  ITEM_LIST                                    ; 1017
                                           0C   AE  9F 00229        PUSHAB  ACMODE
                                           F8   AD  9F 0022C        PUSHAB  NAME_DESC
                                07      0000G   CF  E9 0022F        BLBC    MOUNT_OPTIONS+1, 14$
                                50      0000'   CF  9E 00234        MOVAB   SYSTEM_TABLE, R0
                                                12  11 00239        BRB     16$
                                        0000G   CF  95 0023B 14$:   TSTB    MOUNT_OPTIONS
                                                07  18 0023F        BGEQ    15$
                                50       0088   CE  9E 00241        MOVAB   GROUP_TABLE, R0
                                                05  11 00246        BRB     16$
                                50      0000'   CF  9E 00248 15$:   MOVAB   JOB_TABLE, R0
                                                50  DD 0024D 16$:   PUSHL   R0
                                                7E  D4 0024F        CLRL    -(SP)
                    00000000G   00              05  FB 00251        CALLS   #5, SYS$CRELNM
                                50      0000G   CF  D0 00258        MOVL    MTL_ENTRY, R0                                ; 1021
                           0C   A0              57  D0 0025D        MOVL    R7, 12(R0)
                        0000G   CF              00  FB 00261        CALLS   #0, LOCK_IODB                                ; 1022
                                        0000G   CF  95 00266        TSTB    MOUNT_OPTIONS                                ; 1024
                                                05  19 0026A        BLSS    17$
                                09      0000G   CF  E9 0026C        BLBC    MOUNT_OPTIONS+1, 18$
                                58 00000000G    00  9E 00271 17$:   MOVAB   IOC$GQ_MOUNTLST+4, MOUNT_LIST                ; 1025
                                                10  11 00278        BRB     19$
                                50 00000000G    00  D0 0027A 18$:   MOVL    SCH$GL_CURPCB, R0                            ; 1028
                                5B       0080   C0  D0 00281        MOVL    128(R0), JIB
                                58         04   AB  9E 00286        MOVAB   4(R11), MOUNT_LIST                           ; 1029
                           00   B8      0000G   DF  0E 0028A 19$:   INSQUE  @MTL_ENTRY, @0(MOUNT_LIST)                   ; 1031
                        0000G   CF              00  FB 00290        CALLS   #0, UNLOCK_IODB                              ; 1033
                                        0000G   CF  D5 00295        TSTL    $MTL_ENTRY                                   ; 1038
                                                03  12 00299        BNEQ    20$
                                              01ED  31 0029B        BRW     35$
      FEF8      CD      0000G   DF      0000G   CF  28 0029E 20$:   MOVC3   LOG_NAME, @LOG_NAME+4, LOG_BUFFER            ; 1047
                           F8   AD      0000G   CF  3C 002A8        MOVZWL  LOG_NAME, NAME_DESC                          ; 1048
                           FC   AD       FEF8   CD  9E 002AE        MOVAB   LOG_BUFFER, NAME_DESC+4                      ; 1051
                56      0000G   CF              05  E0 002B4        BBS     #5, MOUNT_OPTIONS+3, 23$                     ; 1053
```

```
                                                         E 11
MAKLOG                                                   16-Sep-1984 01:16:19    VAX-11 Bliss-32 V4.0-742                 Page  19
V04-000                                                  14-Sep-1984 12:45:22    DISK$VMSMASTER:[MOUNT.SRC]MAKLOG.B32;1    (3)

                       07     0000G  CF           05  E1 002BA          BBC      #5, DEVICE_CHAR, 21$                       ; 1056
                                     6E    0000'  CF  9E 002C0          MOVAB    TAPE_PREFIX, P                             ; 1057
                                                  05  11 002C5          BRB      22$                                        ;
                                     6E    0000'  CF  9E 002C7  21$:    MOVAB    DISK_PREFIX, P                             ; 1058
                                     50       08  AC  D0 002CC  22$:    MOVL     VCB, R0                                    ; 1060
                                     59       20  A0  D0 002D0          MOVL     32(R0), RVT                                ;
                                              0C  A9  9F 002D4          PUSHAB   12(RVT)                                    ; 1061
                                                  0C  DD 002D7          PUSHL    #12                                        ;
                              0000V  CF           02  FB 002D9          CALLS    #2, LABEL_LENGTH                           ;
                                 04  AE           50  D0 002DE          MOVL     R0, C                                      ;
                       50        04  AE           05  C1 002E2          ADDL3    #5, C, R0                                  ; 1062
                                 F8  AD           50  B0 002E7          MOVW     R0, NAME_DESC                              ;
                                 FC  AD    FEF8   CD  9E 002EB          MOVAB    LOG_BUFFER, NAME_DESC+4                    ; 1063
                                     5A           50  D0 002F1          MOVL     R0, R10                                    ; 1064
                                     57    FEF8   CD  9E 002F4          MOVAB    LOG_BUFFER, R7                             ;
          5A           00        00  BE           05  2C 002F9          MOVC5    #5, @P, #0, R10, (R7)                      ;
                                                  67     002FF                                                              ;
                                                  0E  18 00300          BGEQ     23$                                        ;
                                     57           05  C0 00302          ADDL2    #5, R7                                     ;
                                     5A           05  C2 00305          SUBL2    #5, R10                                    ;
          5A           00        0C  A9       04  AE  2C 00308          MOVC5    C, 12(RVT), #0, R10, (R7)                  ;
                                                  67     0030F                                                              ;
                                     56    0000G  CF  D0 00310  23$:    MOVL     DEVICE_INDEX, INDEX                        ; 1071
                                     52       04  AC  D0 00315          MOVL     UCB, R2                                    ; 1072
                       02        38  A2           05  E1 00319          BBC      #5, 56(R2), 24$                            ;
                                                  56  D4 0031E          CLRL     INDEX                                      ; 1074
                               00C0  CE 00090004  8F  D0 00320  24$:    MOVL     #589828, ITEM_LIST                         ; 1078
               00C4    CE     0000G  CF           10  C1 00329          ADDL3    #16, SMTL_ENTRY, ITEM_LIST+4               ; 1079
                                             00C8 CE  D4 00331          CLRL     ITEM_LIST+8                                ; 1080
                               00CC  CE 00010004  8F  D0 00335          MOVL     #65540, ITEM_LIST+12                       ; 1087
                               00D0  CE    0000'  CF  9E 0033E          MOVAB    P.AAP, ITEM_LIST+16                        ; 1088
                                             00D4 CE  D4 00345          CLRL     ITEM_LIST+20                               ; 1089
                               00D8  CE 00020004  8F  D0 00349          MOVL     #131076, ITEM_LIST+24                      ; 1090
                               00DC  CE      00C4 CE  9E 00352          MOVAB    ITEM_LIST+4, ITEM_LIST+28                  ; 1091
                                             00E0 CE  D4 00359          CLRL     ITEM_LIST+32                               ; 1092
                               00E4  CE 00010004  8F  D0 0035D          MOVL     #65540, ITEM_LIST+36                       ; 1096
                               00E8  CE    0000'  CF  9E 00366          MOVAB    P.AAQ, ITEM_LIST+40                        ; 1097
                                             00EC CE  D4 0036D          CLRL     ITEM_LIST+44                               ; 1098
                               00F0  CE 00030004  8F  D0 00371          MOVL     #196612, ITEM_LIST+48                      ; 1100
                       07 00000000G  00          00G  E1 0037A          BBC      S^EXE$V_CONCEALED, EXE$GL_FLAGS, 25$       ; 1101
                                     50    0000'  CF  9E 00382          MOVAB    P.AAR, R0                                  ; 1102
                                                  05  11 00387          BRB      26$                                        ;
                                     50    0000'  CF  9E 00389  25$:    MOVAP    P.AAS, R0                                  ; 1103
                               00F4  CE           50  D0 0038E  26$:    MOVL     R0, ITEM_LIST+52                           ; 1101
                                             00F8 CE  D4 00393          CLRL     ITEM_LIST+56                               ; 1104
                                     56           02  C4 00397          MULL2    #2, R6                                     ; 1110
                                         0000GCF46    DF 0039A          PUSHAL   PHYS_NAME[R6]                              ;
               00B8    CE            9E           01  A3 0039F          SUBW3    #1, @(SP)+, PHYSNAM_DESC                   ;
               00BC    CE     0000GCF46           01  C1 003A5          ADDL3    #1, PHYS_NAME+4[R6], PHYSNAM_DESC+4        ; 1111
                                             00BA CE  B4 003AE          CLRW     PHYSNAM_DESC+2                             ; 1112
                               0094  CE 00E80011  8F  D0 003B2          MOVL     #15204369, DVI_ITEM                        ; 1115
                               0098  CE      00A4 CE  9E 003BB          MOVAB    FULLNAM, DVI_ITEM+4                        ; 1116
                               009C  CE      00FC CE  9E 003C2          MOVAB    ITEM_LIST+60, DVI_ITEM+8                   ; 1117
                                             00A0 CE  D4 003C9          CLRL     DVI_ITEM+12                                ; 1118
                                             00FC CE  D4 003CD          CLRL     ITEM_LIST+60                               ; 1119
                                                  7E  7C 003D1          CLRQ     -(SP)                                      ; 1123
                                                  7E  7C 003D3          CLRQ     -(SP)                                      ;
```

```
                                  00A4  CE  9F 003D5       PUSHAB  DVI_ITEM                            :
                                  00CC  CE  9F 003D9       PUSHAB  PHYSNAM_DESC                        :
                                        7E  7C 003DD       CLRQ    -(SP)                               :
               00000000G  00            08  FB 003DF       CALLS   #8, SYS$GETDVIW                     :
                      5F  8F      00A4  CE  91 003E6       CMPB    FULLNAM, #95                        : 1125
                                        19  12 003EC       BNEQ    27$                                 :
            50      00FC  CE            01  C3 003EE       SUBL3   #1, ITEM_LIST+60, R0                : 1127
      00FC  CE            50  00020000  8F  C9 003F4       BISL3   #131072, R0, ITEM_LIST+60           :
                    0100  CE      00A5  CE  9E 003FE       MOVAB   FULLNAM+1, ITEM_LIST+64             : 1128
                                        0C  11 00405       BRB     28$                                 : 1125
                    00FE  CE            02  88 00407 27$:  BISB2   #2, ITEM_LIST+60                    : 1131
                    0100  CE      00A4  CE  9E 0040C       MOVAB   FULLNAM, ITEM_LIST+64               : 1132
                                  0104  CE  7C 00413 28$:  CLRQ    ITEM_LIST+68                        : 1134
                                  00C0  CE  9F 00417       PUSHAB  ITEM_LIST                           : 1150
                                    0C  AE  9F 0041B       PUSHAB  ACMODE                              :
                                    F8  AD  9F 0041E       PUSHAB  NAME_DESC                           :
                          07     0000G  CF  E9 00421       BLBC    MOUNT_OPTIONS+1, 29$                :
                          50     0000'  CF  9E 00426       MOVAB   SYSTEM_TABLE, R0                    :
                                        12  11 0042B       BRB     31$                                 :
                                 0000G  CF  95 0042D 29$:  TSTB    MOUNT_OPTIONS                       :
                                        07  18 00431       BGEQ    30$                                 :
                          50      0088  CE  9E 00433       MOVAB   GROUP_TABLE, R0                     :
                                        05  11 00438       BRB     31$                                 :
                          50     0000'  CF  9E 0043A 30$:  MOVAB   JOB_TABLE, R0                       :
                                        50  DD 0043F 31$:  PUSHL   R0                                  :
                                        7E  D4 00441       CLRL    -(SP)                               :
               00000000G  00            05  FB 00443       CALLS   #5, SYS$CRELNM                      :
                          50     0000G  CF  D0 0044A       MOVL    SMTL_ENTRY, R0                      : 1152
                      0C  A0            52  D0 0044F       MOVL    R2, 12(R0)                          :
                      0B  A0            01  88 00453       BISB2   #1, 11(R0)                          : 1153
                   0000G  CF            00  FB 00457       CALLS   #0, LOCK_IODB                       : 1155
                                 0000G  CF  95 0045C       TSTB    MOUNT_OPTIONS                       : 1157
                                        05  19 00460       BLSS    32$                                 :
                          09     0000G  CF  E9 00462       BLBC    MOUNT_OPTIONS+1, 33$                :
                          58 00000000G  00  9E 00467 32$:  MOVAB   IOC$GQ_MOUNTLST+4, MOUNT_LIST       : 1158
                                        10  11 0046E       BRB     34$                                 :
                          50 00000000G  00  D0 00470 33$:  MOVL    SCH$GL_CURPCB, R0                   : 1161
                          5B      0080  C0  D0 00477       MOVL    128(R0), JIB                        :
                          58        04  AB  9E 0047C       MOVAB   4(R11), MOUNT_LIST                  : 1162
                      00  B8     0000G  DF  0E 00480 34$:  INSQUE  @SMTL_ENTRY, @0(MOUNT_LIST)         : 1164
                   0000G  CF            00  FB 00486       CALLS   #0, UNLOCK_IODB                     : 1166
                          56 00000000G  00  D0 0048B 35$:  MOVL    SCH$GL_CURPCB, R6                   : 1171
            11        27  A6            03  E0 00492       BBS     #3, 39(R6), 36$                     :
            09 00000000G  00            01  E0 00497       BBS     #1, NSA$GR_ALARMVEC, 36$            : 1172
            01 00000000G  00            01  E0 0049F       BBS     #1, NSA$GR_JOURNVEC, 36$            : 1173
                                            04 004A7       RET                                         :
0060  8F    00            6E            00  2C 004A8 36$:  MOVC5   #0, (SP), #0, #96, ARGLIST          : 1182
                                    0C  AE     004AF                                               :
                          53        04  AC  D0 004B1       MOVL    UCB, R3                             : 1183
                          50        1C  A3  D0 004B5       MOVL    28(R3), ORB                         :
                      0C  AE            14  D0 004B9       MOVL    #20, ARGLIST                        : 1189
                      10  AE  00010008  8F  D0 004BD       MOVL    #65544, ARGLIST+4                   : 1192
            04        27  A6            03  E1 004C5       BBC     #3, 39(R6), 37$                     : 1194
                      14  AE            04  88 004CA       BISB2   #4, ARGLIST+8                       : 1196
            04 00000000G  00            01  E1 004CE 37$:  BBC     #1, NSA$GR_ALARMVEC, 38$            : 1197
                      14  AE            01  88 004D6       BISB2   #1, ARGLIST+8                       : 1199
            04 00000000G  00            01  E1 004DA 38$:  BBC     #1, NSA$GR_JOURNVEC, 39$            : 1200
```

```
                        14   AE             02  88 004E2       BISB2    #2, ARGLIST+8                             ; 1202
                        15   AE             07  90 004E6 39$:  MOVB     #7, ARGLIST+9                             ; 1204
                        18   AE 0002000C    8F  D0 004EA       MOVL     #131084, ARGLIST+12                       ; 1211
                        1C   AE             60  D0 004F2       MOVL     (ORB), ARGLIST+16                         ; 1212
                        20   AE 0001000D    8F  D0 004F6       MOVL     #65549, ARGLIST+20                        ; 1214
                                            51  D4 004FE       CLRL     TEMP_PROT                                 ; 1218
                             06        0B   A0  E9 00500       BLBC     11(ORB), 40$                              ; 1219
                             51        18   A0  3C 00504       MOVZWL   24(ORB), TEMP_PROT                        ; 1221
                                            18  11 00508       BRB      41$                                       ;
    51           04          00        18   A0  F0 0050A 40$:  INSV     24(ORB), #0, #4, TEMP_PROT                ; 1224
    51           04          04        1C   A0  F0 00510       INSV     28(ORB), #4, #4, TEMP_PROT                ; 1225
    51           04          08        20   A0  F0 00516       INSV     32(ORB), #8, #4, TEMP_PROT                ; 1226
    51           04          0C        24   A0  F0 0051C       INSV     36(ORB), #12, #4, TEMP_PROT               ; 1227
                        24   AE             51  D0 00522 41$:  MOVL     TEMP_PROT, ARGLIST+24                     ; 1229
                        28   AE 0002000E    8F  D0 00526       MOVL     #131086, ARGLIST+28                       ; 1231
                        2C   AE     0000G   CF  D0 0052E       MOVL     MOUNT_FLAGS, ARGLIST+32                   ; 1232
                             52        30   AE  9E 00534       MOVAB    ARGLIST+36, R2                            ; 1234
                                000000000G  00  16 00538       JSB      NSA$ARGLST_IMGNAM                         ;
                        3C   AE 00040005    8F  D0 0053E       MOVL     #262149, ARGLIST+48                       ; 1236
                        5F   8F      00A4   CE  91 00546       CMPB     FULLNAM, #95                              ; 1237
                                            04  12 0054C       BNEQ     42$                                       ;
                                     00FC   CE  D6 0054E       INCL     ITEM_LIST+60                              ; 1239
                        40   AE      00FC   CE  D0 00552 42$:  MOVL     ITEM_LIST+60, ARGLIST+52                  ; 1240
                        44   AE      00A4   CE  9E 00558       MOVAB    FULLNAM, ARGLIST+56                       ; 1241
                        48   AE 00040006    8F  D0 0055E       MOVL     #262150, ARGLIST+60                       ; 1243
                        4C   AE        F8   AD  3C 00566       MOVZWL   NAME_DESC, ARGLIST+64                     ; 1244
                        50   AE      FEF8   CD  9E 0056B       MOVAB    LOG_BUFFER, ARGLIST+68                    ; 1245
                        54   AE 00040007    8F  D0 00571       MOVL     #262151, ARGLIST+72                       ; 1247
                             52        08   AC  D0 00579       MOVL     VCB, R2                                   ; 1249
                                       14   A2  9F 0057D       PUSHAB   20(R2)                                    ;
                                            0C  DD 00580       PUSHL    #12                                       ;
                     0000V   CF             02  FB 00582       CALLS    #2, LABEL_LENGTH                          ;
                        58   AE             50  D0 00587       MOVL     R0, ARGLIST+76                            ;
                        5C   AE        14   A2  9E 0058B       MOVAB    20(R2), ARGLIST+80                        ; 1250
                             2B        3B   A3  E8 00590       BLBS     59(R3), 43$                               ; 1259
                                       0E   A2  B5 00594       TSTW     14(R2)                                    ; 1260
                                            26  13 00597       BEQL     43$                                       ;
                        0C   AE             03  C0 00599       ADDL2    #3, ARGLIST                               ; 1263
                                       15   AE  96 0059D       INCB     ARGLIST+9                                 ; 1264
                        60   AE 00040008    8F  D0 005A0       MOVL     #262152, ARGLIST+84                       ; 1265
                             59        20   A2  D0 005A8       MOVL     32(R2), RVT                               ; 1266
                                       0C   A9  9F 005AC       PUSHAB   12(RVT)                                   ; 1268
                                            0C  DD 005AF       PUSHL    #12                                       ;
                     0000V   CF             02  FB 005B1       CALLS    #2, LABEL_LENGTH                          ;
                        64   AE             50  D0 005B6       MOVL     R0, ARGLIST+88                            ;
                        68   AE        0C   A9  9E 005BA       MOVAB    12(RVT), ARGLIST+92                       ; 1269
                000000000G   00        0C   AE  FA 005BF 43$:  CALLG    ARGLIST, NSA$EVENT_AUDIT                  ; 1272
                                            04  04 005C7       RET                                                ; 1276

; Routine Size:  1480 bytes,    Routine Base:  $CODE$ + 0015
```

```
1 ROUTINE LABEL_LENGTH (STR_LENGTH, STR_TEXT) =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine will return the length of a given string.
1 !       Trailing blanks at the end of the string are not counted
1 !       as part of the string.
1 !
1 !       NOTE THAT NO VOLUME MAY HAVE A VOLUME LABEL WITH TRAILING BLANKS.
1 !
1 !
1 ! CALLING SEQUENCE:
1 !       LABEL_LENGTH (ARG1, ARG2)
1 !
1 ! INPUT PARAMETERS:
1 !       ARG1: Input string length
1 !       ARG2: Input string address
1 !
1 ! IMPLICIT INPUTS:
1 !       NONE
1 !
1 ! OUTPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !       NONE
1 !
1 !--
1
2 BEGIN
2
2 MAP
2       STR_TEXT        : REF VECTOR [,BYTE];   ! Input string
2
2 LOCAL
2       PTR             : LONG;                 ! Pointer to current char.
2
2 ! Starting at the end of the string, decrement the string length
2 ! until a nonblank character is found, or the beginning of the string
2 ! is encountered.
2 !
2
2 PTR = .STR_LENGTH;
2 WHILE (.PTR GTR 0) AND (.STR_TEXT [.PTR-1] EQL %ASCII' ') DO
2     PTR = .PTR - 1;
2
3 RETURN (.PTR)
1 END;
```

```
MAKLOG                                                              I 11
V04-000                                                  16-Sep-1984 01:16:19    VAX-11 Bliss-32 V4.0-742             Page  23
                                                         14-Sep-1984 12:45:22    DISK$VMSMASTER:[MOUNT.SRC]MAKLOG.B32;1    (4)


                                           0000 00000 LABEL_LENGTH:
                                                                 .WORD    Save nothing                                ; 1277
                              51        04   AC  D0 00002        MOVL     STR_LENGTH, PTR                             ; 1327
                                             0F  15 00006 1$:    BLEQ     2$                                          ; 1328
                   50         51        08   AC  C1 00008        ADDL3    STR_TEXT, PTR, R0                           ;
                              20        FF   A0  91 0000D        CMPB     -1(R0), #32                                 ;
                                             04  12 00011        BNEQ     2$                                          ;
                                             51  D7 00013        DECL     PTR                                         ; 1329
                                             EF  11 00015        BRB      1$                                          ;
                              50             51  D0 00017 2$:    MOVL     PTR, R0                                     ; 1331
                                                 04 0001A        RET                                                  ; 1332

; Routine Size:  27 bytes,    Routine Base:  $CODE$ + 05DD


;  805           1333  1
;  806           1334  1 END
;  807           1335  0 ELUDOM




;                             PSECT SUMMARY
;
;       Name                       Bytes                      Attributes
;
;  $CODE$                            1528  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $PLIT$                             132  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)




;                         Library Statistics
;
;                                            -------- Symbols --------       Pages       Processing
;       File                                 Total    Loaded   Percent       Mapped      Time
;
;  _$255$DUA28:[SYSLIB]LIB.L32;1             18619        94         0       1000          00:01.9




;                               COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:MAKLOG/OBJ=OBJ$:MAKLOG MSRC$:MAKLOG/UPDATE=(ENH$:MAKLOG)

; Size:          1528 code + 132 data bytes
; Run Time:         00:33.7
; Elapsed Time:     01:07.0
; Lines/CPU Min:     2376
; Lexemes/CPU-Min: 26826
; Memory Used:  345 pages
; Compilation Complete
```

0244 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

CHKHM2
LIS

CHNUCB
LIS

GETUIC
LIS

ERASE
LIS

LEFTONE
LIS

MOUDK2
LIS

CHKHM1
LIS

CHKSM2
LIS

CLUSTRMNT
LIS

INIFC2
LIS

MOUDK1
LIS

MAKRVT
LIS

MAKLOG
LIS